

浅析Java语言中的内部类 PDF转换可能丢失图片或格式，建议
阅读原文

https://www.100test.com/kao_ti2020/252/2021_2022__E6_B5_85_E6_9E_90Java_c104_252445.htm 提起Java内部类（Inner Class）可能很多人不太熟悉，实际上类似的概念在C里也有，那就是嵌套类（Nested Class），关于这两者的区别与联系，在下文中会有对比。内部类从表面上看，就是在类中又定义了一个类（下文会看到，内部类可以在很多地方定义），而实际上并没有那么简单，乍看上去内部类似乎有些多余，它的用处对于初学者来说可能并不是那么显著，但是随着对它的深入了解，你会发现Java的设计者在内部类身上的确是用心良苦。学会使用内部类，是掌握Java高级编程的一部分，它可以让你更优雅地设计你的程序结构。下面从以下几个方面来介绍：在这个例子里类Content和GDestination被定义在了类Goods内部，并且分别有着protected和private修饰符来控制访问级别。Content代表着Goods的内容，而GDestination代表着Goods的目的地。它们分别实现了两个接口Content和Destination。在后面的main方法里，直接用 Contents c和Destination d进行操作，你甚至连这两个内部类的名字都没有看见！这样，内部类的第一个好处就体现出来了隐藏你不想让别人知道的操作，也即封装性。同时，我们也发现了在外部类作用范围之外得到内部类对象的第一个方法，那就是利用其外部类的方法创建并返回。上例中的cont()和dest()方法就是这么做的。那么还有没有别的方法呢？当然有，其语法格式如下

```
: outerObject=new outerClass(Constructor  
Parameters).outerClass.innerClass innerObject=outerObject.new
```

InnerClass(Constructor Parameters). 注意在创建非静态内部类对象时，一定要先创建起相应的外部类对象。至于原因，也就引出了我们下一个话题非静态内部类对象有着指向其外部类对象的引用，对刚才的例子稍作修改：`public class Goods { private valueRate=2. private class Content implements Contents { private int i = 11*valueRate. public int value() { return i. } } protected class GDestination implements Destination { private String label. private GDestination(String whereTo) { label = whereTo. } public String readLabel() { return label. } } public Destination dest(String s) { return new GDestination(s). } public Contents cont() { return new Content(). } }`修改的部分用蓝色显示了。在这里我们给Goods类增加了一个private成员变量valueRate，意义是货物的价值系数，在内部类Content的方法value()计算价值时把它乘上。我们发现，value()可以访问valueRate，这也是内部类的第二个好处一个内部类对象可以访问创建它的外部类对象的内容，甚至包括私有变量！这是一个非常有用的特性，为我们在设计时提供了更多的思路和捷径。要想实现这个功能，内部类对象就必须有指向外部类对象的引用。Java编译器在创建内部类对象时，隐式的把其外部类对象的引用也传了进去并一直保存着。这样就使得内部类对象始终可以访问其外部类对象，同时这也是为什么在外部类作用范围之外向要创建内部类对象必须先创建其外部类对象的原因。有人会问，如果内部类里的一个成员变量与外部类的一个成员变量同名，也即外部类的同名成员变量被屏蔽了，怎么办？没事，Java里用如下格式表达外部类的引用：`outerClass.this`有了它，我们就不怕这种屏蔽的情况了。静态内部类和普通的类一样，内部类也可

以有静态的。不过和非静态内部类相比，区别就在于静态内部类没有了指向外部的引用。这实际上和C中的嵌套类很相像了，Java内部类与C嵌套类最大的不同就在于是否有指向外部的引用这一点上，当然从设计的角度以及以它一些细节来讲还有区别。除此之外，在任何非静态内部类中，都不能有静态数据，静态方法或者又一个静态内部类（内部类的嵌套可以不止一层）。不过静态内部类中却可以拥有这一切。这也算是两者的第二个区别吧。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com